**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

| | | | |
|---|---|---|---|
| Application No.: 09/660,563 | § | Examiner: | Lesniewski, Victor D. |
| Filed: September, 12, 2000 | § | Group/Art Unit: | 2153 |
| Inventor: Gregory L. Slaughter, et al. | § | Atty. Dkt. No: | 5181-64900 P4980 |
| Title: MECHANISM AND APPARATUS FOR ACCESSING AND ADDRESSING SERVICES IN A DISTRIBUTED COMPUTING ENVIRONMENT | § | | |

## REPLY BRIEF

**Mail Stop Appeal Brief - Patents**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This brief is in reply to the Examiner's Answer dated May 4, 2007. Appellants respectfully request that this Reply Brief be entered pursuant to 37 C.F.R. § 41.41 and considered by the Board of Patent Appeals and Interferences.

## REPLY

**First Ground of Rejection:**

Claims 1-5, 7-15, 17-25 and 27-30 are rejected under 35 U.S.C. § 102(e) as being anticipated by Beck et al. (U.S. Patent No. 6,604,140). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

**Claims 1-3, 7-9, 11-13, 17 - 19, 21-23 and 27-29:**

**Regarding claim 1, Beck fails to disclose a method comprising a client reading an advertisement from a space, where the space comprises a network-addressable storage location, wherein the advertisement comprises a Uniform Resource Identifier (URI) and a schema, wherein the URI specifies a network address at which a service may be accessed, and wherein the schema specifies one or more messages usable to invoke one or more functions of the service.** What Beck actually teaches is a service framework that enables devices to discover and use services over a network. Beck's service discovery is based on periodic multicasting of service descriptors. Beck teaches that an advertiser retrieves a service to advertise, creates a service descriptor and periodically broadcasts the descriptor (Beck, column 3, lines 59-64 and column 4, lines 40-50).

The Examiner cites, both in the rejection of claim 1 and in the Response to Arguments, column 6, lines 1-16 where Beck describes how a client requests usage of a service by querying a service registry. **However, Beck does not, either at the cited passage or elsewhere, mention a *client reading an advertisement from a space*, as recited in claim 1.** Instead, Beck teaches at the cited passage that a client furnishes a description of the requested service and that the registry matches this request against service descriptors of known services. If a service descriptor in the registry matches the description of the requested service, the registry verifies that the service is already

loaded, and if not, loads the service. Beck does not teach that registry returns to the client (or that the client downloads) the service descriptor, which the Examiner contends is an advertisement. The client in Beck clearly does not read this service descriptor from the registry. Instead, Beck teaches, "[t]he process of binding a service *terminates* in step 507 where a *reference to the service adaptor is returned* to the client" (emphasis added, Beck, column 6, lines 22-24). Elsewhere, Beck teaches that a service adaptor "provides an additional level of indirection between clients and the service" and that the service adaptor is a Java class (Beck, column 5, lines 55-61).

**In the Examiner's Answer** to the Appellants' Appeal Brief, the Examiner continues to maintain that Beck "does disclose a client reading an advertisement from a space", and asserts "[t]he rejection clearly sets forth Beck's service descriptor as an advertisement read from a space (previously cited column 6, lines 1-16)." Col. 6, lines 6-15 are in a description of **FIG. 5**, in which Beck "shows an example of a method of service lookup and binding" via a flowchart. The Figure and accompanying description make it clear that the Examiner's basis for rejection is incorrect. Col. 6, lines 4-8 of Beck state "In step 501, a client requests usage of a service by querying the service registry. In the present embodiment, the client furnishes a description of the requested service via attributes of the service interface and, optionally, the service implementation. The registry matches this request against descriptors of known services." The remaining portion of step 501 and steps 502-507 are then performed by the service registry. The client is not involved in the method illustrated in and described for FIG. 5 after requesting usage of a service by querying the service registry to initiate step 501. Specifically, the client is **not involved in Col. 6, lines 8-21**. Everything described in the relevant portion of the selection is **performed by the service registry, not the client**. The method terminates in step 507 where a reference to the service adapter is returned to the client. In FIG. 5, the client only queries the service registry at the beginning of the description, and presumably would receive the reference to the service adapter (which is clearly **not** the service description itself) returned by the service registry.

In the Examiner's Answer, the Examiner asserts "In Beck's system, the client requests a service by querying a service registry in order to match a certain service descriptor." However, as noted above, the **service registry** matches the client's request to a service descriptor, not the client. Col. 5, lines 9-10 make this point clear: "The **registry** matches this request against descriptors of known services." Appellants note that "requesting a service" is clearly not the same as "reading a service description from a space."

In the Examiner's Answer, the Examiner further asserts "The client reads a matched service descriptor in order to ascertain whether the service needs to be loaded and also in order to download the functionality. This clearly meets the limitation of reading an advertisement from a space." This assertion is clearly incorrect. **In FIG. 5, Beck clearly does not disclose the client reading a service descriptor.** Instead, Beck clearly discloses that the **service registry** matches the client's request against descriptors of known services. Furthermore, Beck clearly discloses that the service registry, and not the client, "ascertains whether the service needs to be loaded and also in order to download the functionality". This clearly disclosed in Beck, col. 5, lines 9-16:

> The **registry** matches this request against descriptors of known services. If a service descriptor matches the description of the requested service, the **registry** follows in step 502 where it checks if the service is already loaded on the device. If the service is not loaded on the device, the **service registry** follows steps 503, 504 and 505 in order to respectively download the service interface, adapter and implementation.

In the cited selection from Beck, it is clear that **the service descriptions are opaque to the client**. The **service registry** "reads" the service descriptions, not the client. The **service registry** checks to see if a service is loaded and, if the service is not loaded, the **service registry** loads the service. In FIG. 5, whether the service was already loaded or was not loaded and thus has to be loaded by the service registry, the service registry returns a reference to a service adapter for the service to the client. **The reference to the service adapter is all the client receives in FIG. 5. The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service**

**registry; Beck's service descriptions that are maintained and accessed by Beck's service registry are clearly opaque to Beck's client.** Contrary to the Examiner's assertion, Beck's FIG. 5 and the accompanying description does **not** "meet the limitation of reading an advertisement from a space."

That Beck, in col. 6, lines 6-15, does not disclose that "the client reads a matched service descriptor…" as asserted by the Examiner is further illustrated in Beck's description of FIG. 2. Beck describes creating a service descriptor at col. 4, lines 41-45:

> In step 201, the advertiser retrieves a service that the device wishes to advertise. In the present embodiment, **this operation is implemented by querying the service registry**. In step 202, the advertiser **creates a service descriptor**.

"Querying the service registry" clearly refers to what is illustrated in FIG. 5 and discussed in col. 6, lines 6-15 ("In step 501, a client requests usage of a service by **querying the service registry**."). Note that, after querying the service registry to "retrieve a service", Beck discloses that the advertiser **creates a service descriptor** for the service. Clearly, if, in col. 6, lines 6-15, Beck's "advertiser" had "read" a service descriptor, the "advertiser" would have no need to **create** a service descriptor.

In the Examiner's Answer, the Examiner further asserts that "in an alternate embodiment, Beck discusses discovery of services and teaches 'To discover services, the service user **needs** to **receive** service descriptors multicasted over the ad-hoc network by other devices.' This would also satisfy the limitation of **reading** an advertisement from a space." (Col. 4, line 66 - col. 5, line 1). Appellants note that Beck does not describe the above as being "in an alternate embodiment." Instead, Beck's language clearly indicates that the above is **necessary to** Beck's system: "the service user **needs** to receive service descriptors multicasted over the ad-hoc network by other devices." Appellants

Furthermore, Beck discloses that the above is performed by a "listener 127" at col. 5, lines 1-2. Appellants note that Beck describes the function of the listener as **passive** "listening" for service descriptions multicasted over a network by other devices.

If Beck's "service user" **receives** a multicasted service description for a service, the "service user" knows about (has "discovered") the corresponding service. If Beck's "service user" does **not** receive a multicasted service description for a service, the "service user" is **not aware** of the service description or a corresponding service. **The service user would not know what to read or where to read it from unless the service user had already received a multicasted service description.** Once a broadcasted service description is received, the service user would have no need to read a service description from a "space."

**Furthermore, Beck's "service user" is nowhere described as performing any proactive steps that could be in any way construed as reading an advertisement from a space.** Beck's service user simply **passively** listens for multicasted service descriptions. Multicasted service descriptions may be **received** via Beck's listener. "Reading an advertisement from a space" is **proactive**, not passive. Passively receiving a service descriptor multicasted on a network is not the same as proactively reading a service descriptor from a "space." One of ordinary skill in the art would recognize the differences between **receiving** a multicasted object via a network and **reading** an object from a space.

Furthermore, as the Examiner noted, Beck teaches that "To discover services, the service user **needs to receive** service descriptors multicasted over the ad-hoc network by other devices." As noted, Beck's language clearly indicates that the above is **necessary to** Beck's system. This leaves no room in Beck for a "service user" to **read a service description from a space**. Indeed, by teaching that, to discover services, the service user **needs** to receive service descriptors multicasted over the network by other devices, Beck actually teaches away from the notion of a service user **reading** an advertisement from a space. Beck indicates that a service user does not discover a service by **reading** an advertisement from a space because Beck clearly indicates that, to discover a service, the service user **needs** to receive a multicasted service descriptor from another device.

Furthermore, Beck only discloses two methods for an entity to obtain a service descriptor: 1) via receiving a multicasted service descriptor from another device as indicated above; and 2) by creating a service descriptor, as indicated above. Beck describes creating a service descriptor at col. 4, lines 41-45. **Beck does not disclose a client reading an advertisement from a space, as the Examiner asserts. Moreover, nowhere does Beck disclose anything like an entity obtaining a service descriptor by "reading a service descriptor from a space."**

In the Response to Arguments, the Examiner also asserts that Beck's teachings regarding a service user receiving service descriptors multicast over the ad-hoc network by other device, without citing any particular portion of Beck. **However, Beck teaches that it is the service user, not a client, that listens for and receives broadcast service descriptors and that the service user saves service descriptors in the registry which functions as described above to locate services for requesting clients.** Furthermore, a service user receiving service descriptors that are *broadcast* over the network cannot be considered a *client reading* an advertisement *from a space*. Receiving a descriptor via a network broadcast is not at all the same as a client reading and advertisement from a space comprising a network-addressable storage location.

In the Examiner's Answer, in response to the Appellants' remarks that "the Examiner also asserts that Beck's teachings regarding a service user receiving service descriptors multicast over the ad-hoc network by other device, without citing any particular portion of Beck", the Examiner states that "it is noted that this portion of Beck was previously clearly cited. See again the citation to col. 4, line 61 through column 5, line 37 as cited in the previous paragraph and as cited in paragraph 17 of the final action dated 11/6/2006." Appellants acknowledge that Col. 4, line 66 - col. 5, line 1 of Beck discloses "To discover services, the service user **needs** to **receive** service descriptors multicasted over the ad-hoc network by other devices." However, Appellants have provided arguments in previous Responses and in this Reply Brief that traverse the Examiner's rejections based on this cited reference.

In the Examiner's Answer, in response to the Appellants' remarks that "Beck teaches that it is the service user, not a client, that listens for and receives broadcast service descriptors", the Examiner asserts that "Beck clearly states that the role of a service user is to host one or more clients that use one or more services on the device. See again column 4, line 61 through column 5, line 37. This relates directly to claim language as clients request usage of services by querying the service registry which, in turn, matches the request against [service] descriptors. Again see column 6, lines 1-16." Appellants acknowledge that Beck does disclose that the role of a service user is to host one or more clients that use one or more services on the device at col. 4, lines 61-63. However, Beck clearly distinguishes between **clients** on the service user and Beck's **service registry** and **listener**, which are contained within Beck's "Service Framework" (see, e.g., FIG. 1).

Beck clearly distinguishes between **clients** on the service user and Beck's **service registry**. As noted above, the **service registry** matches the client's request to a service descriptor, not the client. Col. 5, lines 9-10 make this point clear: "The **registry** matches this request against descriptors of known services." Appellants note that "requesting a service" is clearly not the same as "reading a service description from a space." Beck does not teach or suggest that the client "reads an advertisement from a space", as asserted by the Examiner. Furthermore, Beck clearly distinguishes between **clients** on the service user and Beck's **listener**, which Beck clearly describes as performing the "listening" for multicasted service descriptors on a network. **Neither Beck's "service user" nor Beck's "listener" that is part of Beck's "Service Framework" are described as performing any proactive steps that could be in any way construed as reading an advertisement from a space.** Beck's listener simply **passively** listens for multicasted service descriptions. Multicasted service descriptions may be **received** on a service user via Beck's listener. "Reading an advertisement from a space" is **proactive**, not passive. Passively receiving a service descriptor multicasted on a network is clearly not the same as proactively reading a service descriptor from a "space."

Beck's **listener** and **service registry** are parts of Beck's "Service Framework", and are clearly and distinctly different entities from Beck's **clients**. **Beck does not disclose a client reading an advertisement from a space, as the Examiner asserts. Moreover, nowhere does Beck disclose anything like an entity obtaining a service descriptor by "reading a service descriptor from a space."**

The Examiner also asserted in the Response to Arguments that the above "discussion of the service adaptor … is irrelevant as it can be seen that Beck's client reads an advertisement from a space before downloading of the service interface, adaptor, and implementation." In the Response to Arguments, the Examiner maintains his contention that Beck's teachings regarding a client supplying a description of a requested service and receiving a reference to a service adapter "meets the limitation of 'a client reading an advertisement from a space.'" However, as noted above, the Examiner's interpretation of Beck is simply incorrect. Beck does not teach that a client reads an advertisement from a space before receiving a service adaptor, as the Examiner incorrectly asserts. Instead, as discussed above, Beck's client supplies a service description of a requested service and that the registry finds a matching service to load (if not already loaded). The client is then returned a reference to the service adaptor. Nowhere does Beck's client read an advertisement from a space comprising a network-addressable storage location. A client supplying a description of a requested service and, in return, receiving a reference to a service adaptor, does not disclose a client reading an advertisement from a space comprising a network-addressable storage location, as recited in Appellants' claim. Instead, the client in Beck queries a service registry to obtain a reference to a service adaptor, which cannot be considered an advertisement as defined in claim 1.

In the Examiner's Answer, the Examiner continues to maintain that "Beck goes on to state that 'discovering a service involves loading only a service descriptor, not loading the code that implements the service.' See col. 4, line 61 through col. 5, line 37. The appellants' discussion of the service adapter in support of argument 1 is irrelevant as it can be seen that Beck's client reads an advertisement from a space before downloading

of the service interface, adapter, and implementation. The client reads the service descriptor in order to determine whether this service functionality should be downloaded as discussed above." As Appellants noted above, the Examiner's assertion that "Beck's client reads an advertisement from a space before downloading of the service interface, adapter, and implementation" is clearly incorrect. Beck clearly does not teach that Beck's client "reads an advertisement from a space before downloading of the service interface, adapter, and implementation". **In FIG. 5, Beck clearly does not disclose the client reading a service descriptor from a space.** Instead, Beck clearly discloses that the **service registry** matches the client's request against descriptors of known services. Furthermore, Beck clearly discloses that the service registry, and not the client, "ascertains whether the service needs to be loaded and also in order to download the functionality". This clearly disclosed in Beck, col. 5, lines 9-16:

> The **registry** matches this request against descriptors of known services. If a service descriptor matches the description of the requested service, the **registry** follows in step 502 where it checks if the service is already loaded on the device. If the service is not loaded on the device, the **service registry** follows steps 503, 504 and 505 in order to respectively download the service interface, adapter and implementation.

In the cited selection from Beck, it is clear that **the service descriptions are opaque to the client**. The **service registry** "reads" the service descriptions, not the client. The **service registry** checks to see if a service is loaded and, if the service is not loaded, the **service registry** loads the service. In FIG. 5, whether the service was already loaded or was not loaded and thus has to be loaded by the service registry, the service registry returns only a reference to a service adapter for the service to the client. **The reference to the service adapter is all the client receives in FIG. 5. The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service registry; Beck's service descriptions that are maintained and accessed by Beck's service registry are clearly opaque to Beck's client.**

In the Examiner's Answer, the Examiner further asserts that "the Appellant goes on to state that 'Beck does not teach that a client reads an advertisement from a space

before receiving a service adaptor,' but them immediately contradicts himself by stating 'Beck's client supplies a service description of a requested service and that the registry finds a matching service to load… The client is then returned a reference to the service adaptor.'" Contrary to the Examiner's assertion, the two statements from Appellants' arguments do **not** contradict each other. As Appellants have clearly shown above, Beck does **not** teach that a client reads an advertisement from a space, as the first statement asserts. Appellants' second statement refers to one of Appellants' arguments traversing Examiner's assertion that Beck does so teach. Requesting a service and receiving in response a reference to a service, as is disclosed by Beck in col. 6, lines 3-23, is clearly **not** the same as "reading a service description from a space." In col. 6, lines 3-23, it is clear that **the service descriptions in the registry are opaque to the client**. The **service registry** "reads" the service descriptions, not the client. The **service registry** checks to see if a service is loaded and, if the service is not loaded, the **service registry** loads the service. In FIG. 5, whether the service was already loaded or was not loaded and thus has to be loaded by the service registry, the service registry returns a reference to a service adapter for the service to the client. **The reference to the service adapter is all the client receives in FIG. 5. The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service registry; Beck's service descriptions that are maintained and accessed by Beck's service registry are clearly opaque to Beck's client.** Contrary to the Examiner's assertion, Beck's FIG. 5 and the accompanying description does **not** "meet the limitation of reading an advertisement from a space." Appellants' second statement is **completely consistent** with Appellants' first statement.

In the Examiner's Answer, the Examiner goes on to assert "This in fact meets the limitation at hand as the reference to the service adapter is in fact a network address at which the service may be accessed. Although the client is returned a reference to the service adapter and not the descriptor itself, the descriptor (advertisement) has been read as it has been used to match a service to the request and return a reference to that service. Nowhere do the claims state that the advertisement is downloaded to a client. Claim 1

states only "a client reading an advertisement from a space." Claim 1, when considered as a whole, clearly indicates that, in "reading an advertisement", the client **obtains** the advertisement, as the advertisement, after said "reading", is clearly **not opaque** to the client recited in claim 1. Claim 1 recites:

> a client reading an advertisement from a space, wherein the space comprises a network-addressable storage location, wherein the advertisement comprises a Uniform Resource Identifier (URI) and a schema, wherein the URI specifies a network address at which a service may be accessed, and wherein the schema specifies one or more messages usable to invoke one or more functions of the service; and
>
> the client sending a first message to the service at the URI, wherein the first message is specified in the schema.

After said "reading an advertisement", claim 1 recites that the client sends a first message to the service at the URI, wherein the first message is specified in the schema. Clearly, the client of claim 1 has **obtained** the advertisement, as the client sends a message as specified in the schema comprised in the advertisement to a URI comprised in the advertisement. **This would not be possible if the client of claim 1 did not, in "reading" the advertisement, obtain the actual advertisement.** Again, in col. 6, lines 3-23, Beck's "service descriptions" are not read, obtained, seen, or received by Beck's clients. Beck's "service descriptions" in col. 6, lines 3-23 are opaque to Beck's clients. In contrast, claim 1, when considered as a whole, the "read" advertisements are clearly not opaque to the client. In "reading an advertisement", the client of claim 1 has clearly **obtained** the advertisement.

**Further in regard to claim 1, Beck fails to disclose that the advertisement comprises a Uniform Resource Identifier (URI) and a schema, where the URI specifies a network address at which a service may be accessed, and where the schema specifies one or more messages usable to invoke one or more functions of the service.** The Examiner cites column 4, lines 40-60 of Beck. However, the cited passage does not describe an advertisement that includes a schema that specifies messages usable to invoke functions of a service. Beck teaches that a "service descriptor contains information about the service, including the service name and a description of its function." Beck also states that an "enhanced service descriptor is a service descriptor

that also contains the location of the code implementing the service." (Beck, column 4, lines 45-50). First of all, as discussed above, the service descriptor in Beck is not read by a client from a space comprising a network-addressable storage location. Moreover, Beck does not describe its service descriptor as including *a schema specifying messages usable to invoke functions of the service*. Instead, Beck teaches that a client uses a Java interface for a service to call the methods that the service provides (Beck, column 5, lines 42-46 and column 6, lines 29-36). Beck specifically teaches that a client calls a method provided by the service's interface. Thus, not only does Beck fail to disclose an advertisement that includes a schema specifying messages usable to invoke functions of the service, Beck describes a separate Java interface for the service that a "defines the set of operations that the service can perform on behalf of a client" (Beck, column 5, lines 42-43). The service descriptor, which the Examiner equates to the advertisement of Appellants' claim, clearly does not include a schema specifying messages usable to invoke functions of a service. Moreover, as noted above, Beck's service descriptor cannot be equated to the advertisement of claim 1 because it is not read by a client from a space comprising a network-addressable storage location, as is explicitly stated in Beck.

In the Response to Arguments, the Examiner again cites column 4, lines 40-60 of Beck and asserts that Beck's "service descriptor clearly includes code to allow data transfer between the client and service which effectuates download of service functionalities and thus is 'usable to invoke one or more functions of the service'". Thus, the Examiner's argument appears to be Beck's invoking of service functions somehow discloses the specific limitation of claim 1. The Examiner is ignoring the specific requirements of Appellants' claim. As noted above, Beck teaches the use of a Java Interface that "defines the set of operations that the service can perform". Also as noted above, Beck's Java Interface is clearly separate code and not part of the service descriptor, which the Examiner equates to the advertisement of Appellants' claim. Thus, Beck fails to disclose an advertisement that includes a schema specifying messages usable to invoke functions of the service.

The Examiner also contends that since Beck's service descriptor "leads to the downloading of [the Java] interface, the service descriptor contains code for data transport that is 'usable to invoke one or more functions of the service'". First, the Examiner's statement is not supported by the reference. Nowhere does Beck state that the service descriptor "contains code for data transport that is usable to invoke one or more functions of the service." There is no description in Beck of the service descriptor including a message schema. The Examiner's argument appears to be that since Beck's service descriptor is somehow associated with the service interface, that the service descriptor must also include a schema specifying messages usable to invoke functions of the service. However, a client in Beck's system does not download the service descriptor. Instead, as noted above, the client supplies the service descriptor in order to locate a matching service. Thus, the Examiner's comments regarding Beck's service descriptor have no bearing on Beck's failure to disclose a client reading an advertisement from a space, where the advertisement comprises a schema that specifies one or more messages usable to invoke one or more functions of the service. Additionally, as discussed above, Beck teaches that a "service descriptor contains information about the service, including the service name and a description of its function." Beck also states that an "enhanced service descriptor is a service descriptor that also contains the location of the code implementing the service." Beck does not state that the service descriptor includes a message schema. **Beck clearly fails to mention anything about the service descriptor including a schema specifying messages, and the Examiner's contention that since service descriptor "leads to the downloading" of a Java interface for the service, the service descriptor "contains code for data transfer" is clearly incorrect and unsupported by the true teachings of the reference.**

In the Examiner's Answer, the Examiner maintains that "Beck does disclose an advertisement comprising a schema, wherein the schema specifies one or more messages. The rejection clearly sets forth an enhanced service descriptor that contains information about the service, including the service name, a description, and a location of code (previously cited col. 4, lines 40-60)." Beck, col. 4, lines 45-49 discloses the contents of a "service descriptor":

> The service descriptor contains information about the service, including the service name and a description of its function. An enhanced service descriptor is a service descriptor that also contains the location of the code implementing the service.

A "service name" is clearly not a *schema that specifies one or more messages usable to invoke one or more functions of the service*, as recited in claim 1. A "description of the service's function" is clearly not a *schema that specifies one or more messages usable to invoke one or more functions of the service*, as recited in claim 1. A "location of the code implementing the service" is clearly not a *schema that specifies one or more messages usable to invoke one or more functions of the service*, as recited in claim 1.

In the Examiner's Answer, the Examiner goes on to assert "In Beck's system, the client requests a service by querying a service registry in order to match a certain service descriptor. The client uses a matched service descriptor to download the service functionality. This download includes the service interface, the service adapter, and the service implementation which clearly effectuate the functions of the service." As the Appellants have shown above, the Examiner's assertion that "the client uses a matched service descriptor to download the service functionality" is clearly incorrect. **Beck's service descriptions are opaque to Beck's client**. Beck's **service registry** "reads" the service descriptions, not Beck's client. Beck's **service registry** checks to see if a service is loaded and, if the service is not loaded, Beck's **service registry** loads the service, which includes loading a service interface, a service adapter, and a service implementation. As shown above, Beck's service registry is clearly not Beck's client. As shown above, Beck's client **only** receives a reference to the service adapter. Beck's service description is **opaque** to Beck's client. **The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service registry.**

In the Examiner's Answer, the Examiner goes on to assert "This clearly meets the limitation of an advertisement comprising a schema, wherein the schema specifies

one or more messages usable to invoke one or more functions of the service." Appellants strongly disagree. If a service corresponding to a matched service description is not loaded, Beck's service registry loads the service. Beck's "service description" at best comprises some information related to downloading a corresponding service. As noted above, Beck describes an "enhanced" service descriptor as containing a "service name", a "description of the service's function", and a "location of the code implementing the service." None of those are anything like a *schema that specifies one or more messages usable to invoke one or more functions of the service*, as recited in claim 1. Beck clearly does **not** teach or suggest that the service descriptor comprises *a* ***schema*** *that specifies* ***one or more messages*** *usable to invoke* ***one or more functions of the service***.

In the Examiner's Answer, the Examiner goes on to assert "The service descriptor clearly includes code to allow data transfer between the client and service which effectuates download of service functionalities and thus is 'usable to invoke one or more functions of the service'. See previously cited column 6, lines 1-16. As the appellant states in support of the argument, 'Beck describes a Java interface for the service that 'defines the set of operations that the service can perform on behalf of a client.'' Again, as the reading of a service descriptor leads to the downloading of this interface, the service descriptor contains code for data transfer that is 'usable to invoke one or more functions of the service.'" Appellants have no idea as to what the Examiner is referring to here. Beck discloses nothing at all that would lead one to conclude that Beck's "service descriptor clearly includes code to allow data transfer between the client and service." As noted above, Beck describes an "enhanced" service descriptor as containing a "service name", a "description of the service's function", and a "location of the code implementing the service." None of those are anything like "code to allow data transfer between the client and service." None of the contents of Beck's service descriptor are even described as anything like "code". Furthermore, as noted above, Beck's **service registry** "reads" the service descriptions, not Beck's client. Beck's **service registry** checks to see if a service is loaded and, if the service is not loaded, Beck's **service registry** loads the service, which includes loading a service interface, a service adapter, and a service implementation. **Beck's client is not involved in the above process as**

**disclosed in column 6, lines 1-16.** As shown above, Beck's service registry is clearly not Beck's client. As shown above, Beck's client **only** receives a reference to the service adapter. Beck's service description is **opaque** to Beck's client. **The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service registry.**

**Furthermore, Beck's service interface, service adapter, and service implementation are clearly not Beck's service descriptor or service description, nor are they "comprised in" Beck's service descriptor/service description. The question at hand is whether, in disclosing a "service descriptor" Beck discloses an *advertisement comprising a schema, wherein the schema specifies one or more messages usable to invoke one or more functions of the service.*** The Examiner's arguments are that Beck's "service descriptors/service descriptions" are equivalent to claim 1's "advertisements", and that Beck's "service descriptors/service descriptions" are "read from a space" by Beck's client and then used to access a service, and that Beck's "service descriptors/service descriptions" comprise a schema as recited in claim 1. Now the Examiner is appealing to Beck's "service interface, service adapter, and service implementation", which are clearly and distinctly different from Beck's service descriptor/service description, and which if not present are downloaded by Beck's service registry, to support the assertion that Beck's "service description" discloses limitations of claim 1. **Beck's "service interface, service adapter, and service implementation" are clearly not Beck's service descriptor, and are clearly not comprised in Beck's service descriptor.** The Examiner's complicated and rather obtuse arguments in this section make it even more clear that Beck's "service descriptor" does not teach or suggest anything like an *advertisement comprising a schema, wherein the schema specifies one or more messages usable to invoke one or more functions of the service.* **The Examiner's arguments related to this section actually serve to make it even more clear as to how claim 1 is not anticipated by Beck.**

In the Examiner's Answer, concerning Appellants' remarks that "The Examiner is ignoring the specific requirements of Applicants' claim", the Examiner goes on to assert "it is noted that the appellant has not pointed out what limitations it is felt are being ignored." The limitation that the Examiner is ignoring is that claim 1 recites *an advertisement comprising a schema that specifies one or more messages usable to invoke one or more functions of the service.* As is made clear above, Beck's "service descriptor" can in no way be construed as disclosing this limitation. **Beck clearly does not teach or suggest that Beck's "service descriptor" comprises *a schema that specifies one or more messages usable to invoke one or more functions of the service.*** The Examiner goes on to assert "The appellant continues by pointing out that 'Beck's Java interface is clearly specific code and not part of the service descriptor,' however, as discussed above, the service descriptor clearly includes code to allow data transfer between the client and service." As Appellants made clear above, the Examiner's assertion that Beck's service descriptor "clearly includes code to allow data transfer between the client and service" is incorrect. What Beck's service descriptor actually includes is described by Appellants above, and Beck clearly does not describe the service descriptor as containing anything like "code to allow data transfer between the client and service." The Examiner goes on to assert "This code is in fact a specification of messages that are used to download and run service functionalities (such as the Java interface). This downloading and running means that the messages have been used 'to invoke one or more functions of the service.'" Beck discloses nothing about any such "messages" in the service descriptors. **The Examiner's assertion that "This code is in fact a specification of messages that are used to download and run service functionalities (such as the Java interface)" appears to be something of the Examiner's own invention, as it is clearly not disclosed by Beck.**

In the Examiner's Answer, the Examiner goes on to assert "a schema is simply a data model. In the case of the claims, it is data that 'specifies one or more messages.' Beck's service descriptor, which includes code to allow the client and service to communicate (i.e. specifying messages), clearly meets the limitation at hand as the schema is not further defined in the claims so as to include any additional functionality."

**As Appellants have pointed out above, the Examiner's assertion that "Beck's service descriptor includes code to allow the client and service to communicate" is incorrect.** The Examiner goes on to assert "It is further noted that the claims state that the messages are what is 'usable to invoke one or more functions of the service.' Again, the communication between the client and service results in the downloading and running of the service interface, the service adaptor, the service implementation, etc." Appellants note that in claim 1 of the instant application, a client simply reads an advertisement from a space to obtain the advertisement itself, and then may invoke one or more functions of the service by sending messages specified in the schema directly to the service at a URI comprised in the schema. **No downloading of a service interface, service implementation, and service adaptor, as disclosed in Beck, are required. The Examiner's assertion that Beck discloses "the communication between the client and service results in the downloading and running of the service interface, the service adaptor, the service implementation, etc.", if anything, actually serves to make it even more clear as to how claim 1 is not anticipated by Beck.**

**Moreover, the Examiner is improperly attempting to equate various aspects of a "service framework for computing devices" as disclosed in various sections of Beck that involves several different and distinct entities (e.g. a client and a service registry) with Appellants' claim 1.** In Beck's "service framework for computing devices", Beck discloses that the client requests a service. A service registry receives the request, attempts to match the request with a service description in the registry, and if a matching service description is found, the service registry checks to see if the corresponding service is loaded. If the service is not loaded, the service registry loads the service. The service registry then returns (**only**) a reference to a service adaptor to the client. The client would then presumably invoke one or more functions of the service via Beck's service interface and service adaptor (**not** via Beck's service descriptor) which were, if necessary, loaded by Beck's service registry, and **not** by Beck's client, as illustrated in FIG. 6 of the Beck reference. In contrast, in claim 1 of the instant application, a client simply reads an advertisement from a space to obtain the advertisement itself, and then may invoke one or more functions of the service by

sending messages specified in the schema directly to the service at a URI comprised in the schema. No loading of a service interface, service implementation, and service adaptor are required. No invocation of a function of the service as illustrated in claim 6 of Beck is required. No service registry is required. **The Examiner's arguments actually serve to make it even more clear as to how claim 1 is not anticipated by Beck.** Beck's "service framework for computing devices" is clearly and distinctly different than what is recited in claim 1 of the instant application. Claim 1 of the instant application is clearly not anticipated by Beck.

**Furthermore, the Examiner has contended both that Beck's client "reads an advertisement from a space" and that Beck's client downloads a service using a "read" service descriptor.** Since the use of the service registry on Beck's "service user" is taught as a key aspect of Beck's "service framework for computing devices", Beck actually appears to teach against Beck's client "reading an advertisement from a space" and Beck's client downloading a service using a "read" service descriptor. Beck's client performing these functionalities would actually **go against** the intent (and the disclosed operation) of Beck's "service framework for computing devices", which in part is to abstract clients on a device from having to perform functions that Beck describes as being performed by Beck's "service framework for computing devices", which includes Beck's service registry.

Anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention** must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Beck clearly fails to disclose a client reading an advertisement from a space, wherein the space comprises a network-addressable storage location, wherein the advertisement comprises a Uniform Resource Identifier (URI) and a schema, wherein the URI specifies a network address at which a service may be accessed,

and wherein the schema specifies one or more messages usable to invoke one or more functions of the service. Therefore, Beck cannot possibly be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art, and removal thereof is respectfully requested.

**Claims 4, 14, and 24:**

**Regarding claim 4, Beck fails to disclose that the schema is expressed in a data representation language.** The Examiner cites column 5, lines 46-50. However, the cited portion of Beck teaches that a service interface 402 defines the set of operations that the service can perform on behalf of a client and that the service interface is a Java interface. As is well known in the art, a Java interface is not a schema *expressed in a data representation language*. Nowhere does Beck mention any schema expressed in a data representation language and clearly fails to disclose a schema expressed in a data representation language that is included in an advertisement. Without some teaching of Beck regarding a schema expressed in a data representation language, Beck cannot be said to anticipate claim 4.

The Examiner, in the Response to Arguments, contends that Appellant, "has failed to provide any basis for [the] conjecture" that a Java Interface is not a schema expressed in a data representation language. **However, it is the Examiner who shoulders the burden of proof, not the Appellants.** The Examiner also asserts that a "data representation language is not further defined or explained in the claims so as to be distinguished over the Java programming language" and that "Java abstracts the data on bytecodes so what when applications are developed the same code may run in different environments." However, the Examiner's statements actually support Appellants' arguments. Java utilized bytecodes, which, by definition, cannot in any way be considered a data representation language. As described in Appellants' specification, and as is well understood by anyone of ordinary skill in the art, a data representation language (such as XML) is a particular type of language used to describe or represent data or

content. No one of ordinary skill in the art would consider Java to be a data representation language.

In response to the Examiner's argument that Appellant has not provided any basis for the fact that a Java interface is not a schema expressed in a data representation language, Appellant submits that it is the Examiner's burden to "make clear that the missing descriptive matter is ***necessarily*** present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill" (emphasis added). As described in the M.P.E.P at 2131.01 (III), "[t]o serve as an anticipation when the reference is silent about the asserted inherence characteristic, such gap in the reference may be filled with recourse to extrinsic evidence." In the present rejection, the Examiner merely cites column 5, lines 46-50 that does not include any mention of a schema expressed in a data representation language. Instead, the cited passage merely describes a Java Interface. The Examiner has not provided any extrinsic evidence that Beck's Java Interface ***necessarily*** includes a **schema** expressed in a data representation language. Instead, the Examiner merely makes conclusory statements regarding the Java programming language, class, JAR files and bytecodes (which clearly cannot be considered either a schema nor expressed in a data representation language). Furthermore, **as is well known in the art**, JAVA interfaces do not include schemas expressed in a data representation language.

**In the Examiner's Answer,** the Examiner simply maintains that "Beck does disclose a schema and a message expressed in a data representation language. The rejection clearly sets forth Beck's use of a Java interface and Java classes (previously cited column 5, lines 46-61)…Again, the use of Java-based schema and messages meet the limitation in question." Appellants note that, in <argument 3> of the Examiner's Answer, **the Examiner has provided nothing substantive in that adds to the arguments the Examiner previously submitted in the Final Office Action**. The Examiner's argument in Answer is virtually identical to the Examiner's arguments in the Final office Action. Therefore, Appellants' above-repeated arguments still stand. **The Examiner has <u>still</u> not provided any extrinsic evidence that Beck's Java Interface**

***necessarily*** **includes a schema expressed in a data representation language. Instead, the Examiner simply repeats conclusory statements regarding the Java programming language, class, JAR files and bytecodes (which clearly cannot be considered either a schema nor expressed in a data representation language). Furthermore, as is well known in the art, JAVA interfaces do not include schemas expressed in a data representation language**.

In the Examiner's Answer, the Examiner goes on to assert "Further, it is contended that the appellant is reading limitations into the claims that are not present as a data representation language is not further defined or explained in the claims. The appellant argues that 'a data representation language is a particular type of language used to describe or represent data or content.' Here it is noted that the Java language meets this definition." The Appellant argues that a data representation language is "a particular type of language used to describe or represent data or content" because that is what a data representation is ***by definition***. Further, the term "data representation language" is well-known in the art, and the Appellants' definition of "data representation language" would be recognized by one of ordinary skill in the art. One of ordinary skill in the art would recognize that Java is not a **data representation language**. One of ordinary skill in the art would not recognize "a data representation language is a particular type of language used to describe or represent data or content" as a definition corresponding to Java. Java would be described and recognized in the art as a **programming language**, not as a **data representation language**.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested.

**Claims 5, 15 and 25:**

**Regarding claim 5, Beck fails to disclose where the first message is expressed in a data representation language.** The Examiner cites column 5, lines 54-61 and column 6, lines 30-39. The cited portions of Beck describe that once a client has bound a

service it can use that service by calling a method provided by the service's interface and that the service interface forwards the call to the service implementation that performs the requested service. Beck also teaches that the service interface and the service implementation are Java-based and that the RMI, OSF-RPC and IIOP inter-process communication protocols are used. Beck does not mention anything about a message expressed in a data representation language (such as XML) and clearly fails to disclose wherein a message send by a client to the service is expressed in a data representation language. The Examiner is clearly speculating (which is improper) regarding the details of Beck's messages. As Beck makes no mention of any message expressed in a data representation language, Beck does not anticipate claim 5. In the Response to Arguments, the Examiner argues that Beck's Java Interface discloses messages expressed in a data representation language. However, as discussed above regarding claim 4, the Examiner's interpretation of Beck is incorrect. Please refer to the remarks above regarding claim 4 regarding the fact that a Java interface does not include or disclose anything expressed in a data representation language.

**In the Examiner's Answer,** the Examiner simply maintains that "Beck does disclose a schema and a message expressed in a data representation language. The rejection clearly sets forth Beck's use of a Java interface and Java classes (previously cited column 5, lines 46-61)…Again, the use of Java-based schema and messages meet the limitation in question." Appellants note that, in <argument 3> of the Examiner's Answer, **the Examiner has provided nothing substantive in that adds to the arguments the Examiner previously submitted in the Final Office Action**. The Examiner's argument in Answer is virtually identical to the Examiner's arguments in the Final office Action. Therefore, Appellants' above-repeated arguments still stand. **The Examiner has <u>still</u> not provided any extrinsic evidence that Beck's Java Interface *necessarily* includes a schema expressed in a data representation language. Instead, the Examiner simply repeats conclusory statements regarding the Java programming language, class, JAR files and bytecodes (which clearly cannot be considered either a schema nor expressed in a data representation language).**

**Furthermore, as is well known in the art, JAVA interfaces do not include schemas expressed in a data representation language**.

In the Examiner's Answer, the Examiner goes on to assert "Further, it is contended that the appellant is reading limitations into the claims that are not present as a data representation language is not further defined or explained in the claims. The appellant argues that 'a data representation language is a particular type of language used to describe or represent data or content.' Here it is noted that the Java language meets this definition." The Appellant argues that a data representation language is "a particular type of language used to describe or represent data or content" because that is what a data representation is ***by definition***. Further, the term "data representation language" is well-known in the art, and the Appellants' definition of "data representation language" would be recognized by one of ordinary skill in the art. One of ordinary skill in the art would recognize that Java is not a **data representation language**. One of ordinary skill in the art would not recognize "a data representation language is a particular type of language used to describe or represent data or content" as a definition corresponding to Java. Java would be described and recognized in the art as a **programming language**, not as a **data representation language**.

For at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 15 and 25.

**Claims 10, 20 and 30:**

Regarding claim 10, Beck fails to disclose the client using the URI and the schema in the advertisement to construct a gate for access to the service. The Examiner cites column 7, lines 34 – 44 of Beck. However, this passage of Beck does not describe a client using the URI and the schema in the advertisement to construct a gate for access to the service. First of all, as noted above regarding the rejection of claim 1, Beck fails to disclose a client reading an advertisement that comprises a schema that specifies

messages usable to invoke functions of the service. Secondly, nowhere does Beck describe the client using a schema from an advertisement to construct a gate for access to the service. Instead, Beck teaches that the service registry downloads the service interface, adapter and implementation and returns a reference to the client and that the client calls methods provided in the service's interface that forwards the call to the service adapter (Beck, column 6, lines 3-24 and lines 29-44). The client does not have to construct anything in Beck, and Beck certainly fails to describe a client using the URI *and the schema* in the advertisement to construct a gate for access to the service.

In the Response to Arguments, the Examiner again cites column 7, lines 34 – 44 where Beck describes a service implementation that split into an implementation proxy and a remote service implementation. As noted above, the cited passage makes no mention of a client using the URI *and the schema in the advertisement* to construct a gate for access to the service. The Examiner is improperly ignoring the specific limitations of claim 10. Specifically, the Examiner is ignoring the fact that claim 10 recites, in part, a "client **using the URI and the schema in the advertisement** to construct a gate for access to the service" (emphasis added). The Examiner has not cited any portion of Beck, nor provided any other evidence or interpretation in support for the contention that Beck's client uses a schema *in the advertisement* to construct a gate to access the service. The Examiner argues that by downloading a service interface, Beck client is somehow using a URI and a schema from an advertisement to construct a gate for access to the service. However, as noted above, Beck's service interface cannot be considered "a schema" and is clearly not included in the service descriptor, which the Examiner equates to the Advertisement of Appellants' claims.

In the Examiner's Answer, the Examiner simply maintains that "Beck does disclose the client using the URI and the schema in the advertisement to construct a gate for access to the service. The rejection clearly sets forth Beck's construction of a gate for access to the service as it cites the use of the service implementation at either side or both sides of the data transfer (previously cited column 7, lines 34-44). As discussed above, in Beck's system, the client requests a service by querying a service registry in order to

match a certain service descriptor. The client uses a matched service descriptor to download the service functionality. This download includes the service interface, the service adapter, and the service implementation which clearly effectuate the functions of the service. See previously cited column 6, lines 1-16. This clearly meets the limitation of using the advertisement to construct a gate for access to the service. Also it is again noted that an enhanced service descriptor contains a URL that is used to access the service. See previously cited column 4, lines 40-60." The Examiner's assertion that Beck discloses "the client uses a matched service descriptor to download the service functionality" is clearly incorrect. Beck clearly discloses that the service registry, and not the client, downloads the service functionality, if necessary. This clearly disclosed in Beck, col. 5, lines 9-16:

> The **registry** matches this request against descriptors of known services. If a service descriptor matches the description of the requested service, the **registry** follows in step 502 where it checks if the service is already loaded on the device. If the service is not loaded on the device, the **service registry** follows steps 503, 504 and 505 in order to respectively download the service interface, adapter and implementation.

In the cited selection from Beck, it is clear that **the service descriptions are opaque to the client**. The **service registry** "reads" the service descriptions, not the client. The **service registry** checks to see if a service is loaded and, if the service is not loaded, the **service registry** loads the service. In FIG. 5, whether the service was already loaded or was not loaded and thus has to be loaded by the service registry, the service registry returns a reference to a service adapter for the service to the client. **The reference to the service adapter is all the client receives in FIG. 5. The client never reads, receives, or sees the service description itself. Beck's service descriptions are maintained and accessed by Beck's service registry and only by Beck's service registry; Beck's service descriptions that are maintained and accessed by Beck's service registry are clearly opaque to Beck's client.**

Appellants maintain that Beck fails to disclose the client using the URI and the schema in the advertisement to construct a gate for access to the service, and further maintain that Beck does not teach or suggest using an URI and a schema in an

advertisement to construct a gate **at all**. However, **even if** Beck disclosed using an URI and a schema in an advertisement to construct a gate, Beck **clearly** teaches that the **service registry**, and not the **client**, loads the service. Since the use of the service registry on Beck's "service user" is taught as a key aspect of Beck's "service framework for computing devices", Beck actually appears to teach against *the **client** using the URI and the schema in the advertisement to construct a gate for access to the service*, as is recited in claim 10. Beck's client "constructing a gate" (or downloading a service) would actually **go against** the intent of Beck's "service framework for computing devices", which in part is to abstract clients on a device from having to perform functions that Beck describes as being performed by Beck's "service framework for computing devices".

Thus, column 6, lines 1-16 of Beck clearly do not meet the limitation of *the **client** using the URI and the schema in the advertisement to construct a gate for access to the service.*

In the Examiner's Answer, the Examiner asserts "Further, it is contended that the appellant is reading limitations into the claims that are not present as constructing a gate is not further defined or explained in the claims except that it leads to accessing the service. Thus, it is maintained that Beck's use of the service implementation at either side or both sides of the data transfer meets the limitation of constructing a gate as it results in accessing the service as described above." Applicants fail to see what the limitations are that the Examiner asserts that Appellant is "reading into the claims". The Examiner only states in regards to this contention "…as constructing a gate is not further defined or explained in the claims except that it leads to accessing the service." However, Appellants' arguments do not attempt to "read limitations into the claims that are not present" in regards to a gate. Appellants' arguments do not depend on any further definition or explanation of a "gate" than what is recited in the claims.

For at least the reasons above, the rejection of claim 10 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks apply to claims 20 and 30.

**Second Ground of Rejection:**

Claims 6, 16, and 26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Beck in view of Roberts et al. (U.S. Patent Number 6,560,633). Appellants traverse this rejection for at least the following reasons.

In the Response to Arguments, the Examiner asserts, "it is maintained that the stated motivation in the above rejection is sufficient". The Examiner's statement that the combination of Beck and Roberts "satisfies the need for a more efficient approach to service discovery that uses more efficient methods of describing and loading services" does not provide any motivation to modify Beck and is not supported by any evidence of record. Beck's entire invention is directed toward service discovery. Thus, one seeking an approach to service discovery would simply use Beck's invention, not modify it, as the Examiner contends.

Furthermore, Beck describes a specific Java interface for the service that a "defines the set of operations that the service can perform on behalf of a client" (Beck, column 5, lines 42-43). To modify Beck with Roberts to use XML messages would be counter to the intended operation of Beck to employ a specific Java interface. If a proposed modification would render the prior art feature unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900 (Fed. Cir. 1984). Accordingly, it would be improper to modify Beck's teachings to employ XML messages.

The Examiner, in the Response to Arguments, asserts "Beck specifically offers language independence in his system to allow the data transfer and other possible service implementations to be written in any programming language." **However, the fact that Beck teaches that service implementation may be written in any *programming* language actually supports Appellants' argument.** As is well understood in the art, XML is a data representation language, not a programming language. Furthermore,

modifying the programming language in which Beck's service implementation is developed does not provide any motivation to express a message specified in a schema in a data representation language that includes XML, as required by the specific limitations in Appellants' claim.

In the Examiner's Answer, the Examiner maintains that "the combination of Beck and Roberts does disclose the data representation language comprising XML. Roberts clearly teaches the use of XML in the context of expressing messages sent to invoke a service." Contrary to the Examiner's assertion, Appellants can find nothing in Roberts that teaches or suggests the use of XML "in the context of expressing messages sent to invoke a service." Nor did the Examiner provide any citations from Roberts in regard to this assertion. However, Roberts does disclose the use of XML, and XML is a data representation language.

In the Examiner's Answer, the Examiner goes on to assert "Thus the combination of Beck and Roberts teaches the limitation in question as discussed in the rejection in section (9) above." Appellants note that claim 6 depends from claims 5 and 1, and Appellants have clearly shown in above arguments that Beck does not anticipate claims 5 and 1 as the Examiner asserts. Thus, on this basis alone, the combination of Beck and Roberts does not teach the limitations of claim 6.

In the Examiner's Answer, the Examiner goes on to assert "Concerning the appellants' statements about motivation to combine the references, it is maintained that the stated motivation in the above rejection is sufficient. See the paragraphs discussing the combination of Beck and Roberts in the rejection in section (9) above." Appellants have responded to this assertion. The Examiner's statement that the combination of Beck and Official Notice "satisfies the need for a more efficient approach to service discovery that uses more efficient methods of describing and loading services" does not provide any motivation to modify Beck and is not supported by any evidence of record. Beck's entire invention is directed toward service discovery. Thus, one seeking an approach to service discovery would simply use Beck's invention, not modify it, as the Examiner contends.

The Examiner has not responded to this argument. Furthermore, the Examiner's stated motivation to combine the references is merely conclusory. Neither Beck nor Roberts disclose any motivation to combine the two references. Specifically, neither Beck nor Roberts disclose any motivation to use XML in Beck's system as the Examiner argues. Furthermore, the Examiner's argument that XML **could** be used in Beck as the Examiner asserts is without merit, as is indicated below.

Appellants remind the Examiner that obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. *In re Bond,* 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). In addition, the showing of a suggestion, teaching, or motivation to combine prior teachings "must be clear and particular .... Broad conclusory statements regarding the teaching of multiple references, standing alone, are not 'evidence'." *In re Dembiczak*, 175 F.3d 994, 50 USPQ2d 1614 (Fed. Cir. 1999). The art must fairly teach or suggest to one to make the specific combination as claimed. That one achieves an improved result by making such a combination is no more than hindsight without an initial suggestion to make the combination. The Examiner has failed to provide a proper *prima facie* case of obviousness.

In the Examiner's Answer, the Examiner goes on to assert "Furthermore, the appellant has stated that "To modify Beck to use XML messages would be counter to the intended operation of Beck to employ a specific Java interface." However, this is incorrect. Beck specifically offers language independence in his system to allow the data transfer and other possible service implementations to be written in any **programming** language. See Beck, column 6, lines 63-65." Appellants have previously responded to this assertion. **The fact that Beck teaches that service implementation may be written in any *programming* language actually supports Appellants' argument.** As is well understood in the art, XML is a **data representation language**, not a **programming language**. One of ordinary skill in the art would easily recognize the distinction between **data representation languages** and **programming languages**. Beck does not teach or suggest, or provide any motivation for, writing service

implementations in **data representation languages**. By definition, a data representation language is "a particular type of language used to describe or represent data or content". Data representation languages are **not** programming languages, are instead used to describe or represent data or content, and are **not** designed or intended for "writing service implementations." Furthermore, modifying the programming language in which Beck's service implementation is developed does not provide any motivation to express a message specified in a schema in a data representation language that includes XML, as required by the specific limitations in Appellants' claim. **The Examiner has not substantively responded to this argument.**

In the Examiner's Answer, the Examiner goes on to assert "Here it is noted that the appellant has made no mention of the teachings of Roberts as cited in the rejection. The appellant is reminded that the rejection is based on the combination of Beck and Roberts, both of which are directed toward expressing messages sent to invoke services." Again, contrary to the Examiner's assertion, Appellants can find nothing in Roberts that teaches or suggests the use of XML "in the context of expressing messages sent to invoke a service." Furthermore, the Examiner has not provide any specific citations from Roberts in regard to this assertion for the Appellants to respond to. However, Roberts does disclose the use of XML, and XML is a data representation language. That Roberts discloses the use of XML, and that XML is a data representation language, appears to be the only thing of substance in regards to Roberts and Examiner's assertion that a combination of Beck and Roberts teaches the limitations of claim 6.

In the Examiner's Answer, the Examiner goes on to assert "The appellant is reminded that one cannot show nonobviousness by attacking references individually where the rejections are based on combination of references. The appellant is again directed to the combination in the rejection above." Appellants have shown that Beck does not anticipate claims 1 and 5, from which claim 6 depends. Roberts does disclose the use of XML, and XML is a data representation language. Furthermore, the Examiner's argument that XML **could** be used in Beck as the Examiner asserts is without merit, as is indicated above. For example, **XML is not a programming language**, and

thus is not usable in Beck as the Examiner asserts. Moreover, Roberts only describes XML in it's traditional use for representing data. Roberts does not teach the use of XML for expressing messages sent to invoke a service, which is not something that XML was designed for.

**Furthermore, even if Beck and Roberts were combinable and were combined, the results would not be anything like what is disclosed in claim 6 of the instant application.** Combining Beck's "Service framework for computing devices" with Roberts' "Method for creating network services by transforming an XML runtime model in response to an iterative input process", if possible, would not produce what is recited in claim 6 of the instant application. The results of such a combination would simply be an embodiment of Beck's "Service framework for computing devices", which does not anticipate what is recited in claims 1 and 5, that uses XML for some purpose.

For at least the reasons above, the rejection of claims 6, 16 and 26 is not supported by the prior art and removal thereof is respectfully requested.

## CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-30 is erroneous, and reversal of the Examiner's decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-64900/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: July 1, 2007